

**Amendments to the Specification**

**Please replace paragraph [0024] with the following re-written paragraph:**

[0024] More specifically, mapping engine 106 may include parser logic, tables and associated resources to detect incoming XAML data in the form of XAML nodes 104 from source XAML 102 or otherwise, and generate a corresponding binary representation 108 of those XAML nodes 104. The binary representation 108 may in one regard encapsulate the information characterizing object tree 110 in binary form. In embodiments, the binary representation 108 may include tokenized decompositions of XAML data, with common assembly, attribute and other information ~~hosted~~ sent to common files to make the representation more compact.

**Please replace paragraph [0033] with the following re-written paragraph:**

[0033] If a type implements serialization interfaces configured to be consistent with the invention, it may be called upon when a stream of binary representation 108 is constructed to serialize itself directly into the stream in its own custom defined format resulting in compact storage and transmission requirements. The Length (or other dimensional) type can exploit this type of capability effectively, resulting in typical cases in a requirement of only 1 byte to specify the length of an object in a binary stream. Fig. 5 for example shows illustrative code for performing a serialization for a Length type. Other implementing code is possible. The binary representation 108 and associated ~~may~~ records may thus likewise be optimized by generating a type index for novel types (such as Length) upon encountering the first instance of those types. Loading times may also therefore be enhanced.

**Please replace paragraph [0034] with the following re-written paragraph:**

[0034] In terms of use of the binary representation 108 once it is generated according to embodiments of the invention, as illustrated in Fig 6 the mapping engine 106 may likewise communicate with a binary reader 116, that module being configured to receive and transmit the binary representation 108 and communicate with a mapping table 134 and other resources. Mapping table 134 may contain lookup tables or other entries storing common attributes, types and other information which may be used to extract ~~hoisted or~~ encoded information from binary representation 108. As shown in Fig. 6, the binary reader 116 may for instance be invoked to read the binary representation 108 and generate a reconstructed object 120 corresponding to object tree 110 or other output. That output may be used for example to expose a dialog box or other user interface element or other object to an application programming interface 124. The application programming interface 124 may interface to applications such as browsers, databases or other client or network applications to call system or other resources. In embodiments, as illustrated the binary reader 116 may also generate a text representation 122 of the object tree 110 for transmission to a text-related application programming interface 136 or other applications or resources.

**Please replace paragraph [0035] with the following re-written paragraph:**

[0035] Fig. 7 illustrates aspects of processing which may be used to generate a converted binary representation of source XAML 102, according to embodiments of the invention. In step 702, processing may begin. In step 704, source XAML 102 may be loaded or opened in mapping engine 106 ~~or otherwise~~. In step 706, the next (or initial) XML token may be converted to a XAML token. In step 708, the XAML token may be

converted to a binary record, for instance as part of the set of binary records 130. In step 710, the binary representation record may be written to the set of binary records 130 or other locations, for instance using binary writer 118. In step 712, a determination may be made whether there is additional XML to be read from source XAML 102 or otherwise. If there is additional XML input to be read, processing may return to step 706 to map the next XML token to a XAML token. If there is no additional XML input to be read, processing may proceed to step 714 where a file containing the set of binary records 130 may be closed. In step 716, processing may repeat, terminate or return to a prior processing point.